Network Working Group
Request for Comments: 3378
Category: Informational

R. Housley
RSA Laboratories
S. Hollenbeck
VeriSign, Inc.

September 2002

EtherIP: Tunneling Ethernet Frames in IP Datagrams

Status of this Memo

Abstract

This document describes the EtherIP, an early tunneling protocol, to provide informational and historical context for the assignment of IP protocol 97. EtherIP tunnels Ethernet and IEEE 802.3 media access control frames in IP datagrams so that non-IP traffic can traverse an IP internet. The protocol is very lightweight, and it does not provide protection against infinite loops.

1. Introduction

EtherIP was first designed and developed in 1991. This document was written to document the protocol for informational purposes and to provide historical context for the assignment of IP protocol 97 by IANA.

The IETF Layer Two Tunneling Protocol Extensions (L2TPEXT) Working Group and IETF Pseudo Wire Emulation Edge-to-Edge (PWE3) Working Group are developing protocols that overcome the deficiencies of EtherIP. In general, the standards track protocols produced by these IETF working groups should be used instead of EtherIP.

The EtherIP protocol is used to tunnel Ethernet [DIX] and IEEE 802.3 [CSMA/CD] media access control (MAC) frames (including IEEE 802.1Q [VLAN] datagrams) across an IP internet. Tunneling is usually performed when the layer three protocol carried in the MAC frames is not IP or when encryption obscures the layer three protocol control information needed for routing. EtherIP may be implemented in an end station to enable tunneling for that single station, or it may be implemented in a bridge-like station to enable tunneling for multiple stations connected to a particular local area network (LAN) segment.

EtherIP may be used to enable communications between stations that implement Ethernet or IEEE 802.3 with a layer three protocol other than IP. For example, two stations connected to different Ethernet LANs using the Xerox Network Systems Internetwork Datagram Protocol (IDP) [XNS] could employ EtherIP to enable communications across the Internet.

EtherIP may be used to enable communications between stations that encrypt the Ethernet or IEEE 802.3 payload. Regardless of the layer three protocol used, encryption obscures the layer three protocol control information, making routing impossible. For example, two stations connected to different Ethernet LANs using IEEE 802.10b [SDE] could employ EtherIP to enable encrypted communications across the Internet.

EtherIP may be implemented in a single station to provide tunneling of Ethernet or IEEE 802.3 frames for either of the reasons stated above. Such implementations require processing rules to determine which MAC frames to tunnel and which MAC frames to ignore. Most often, these processing rules are based on the destination address or the EtherType.

EtherIP may be implemented in a bridge-like station to provide tunneling services for all stations connected to a particular LAN segment. Such implementations promiscuously listen to all of the traffic on the LAN segment, then apply processing rules to determine which MAC frames to tunnel and which MAC frames to ignore. MAC frames that require tunneling are encapsulated with EtherIP and IP, then transmitted to the local IP router for delivery to the bridge-like station serving the remote LAN. Most often, these processing rules are based on the source address, the destination address, or the EtherType. Care in establishing these rules must be exercised to ensure that the same MAC frame does not get transmitted endlessly between several bridge-like stations, especially when broadcast or multicast destination MAC addresses are used as selection criteria. Infinite loops can result if the topology is not restricted to a tree, but the construction of the tree is left to the human that is configuring the bridge-like stations.

1.1. Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Protocol Format

EtherIP segments are sent and received as internet datagrams. An Internet Protocol (IP) header carries several information fields, including an identifier for the next level protocol. An EtherIP header follows the internet header, providing information specific to the EtherIP protocol.

Internet Protocol version 4 (IPv4) [RFC791] defines an 8-bit field called "Protocol" to identify the next level protocol. The value of this field MUST be set to 97 decimal (141 octal, 61 hex) to identify an EtherIP datagram.

EtherIP datagrams contain a 16-bit header and a variable-length encapsulated Ethernet or IEEE 802.3 frame that immediately follows IP fields.

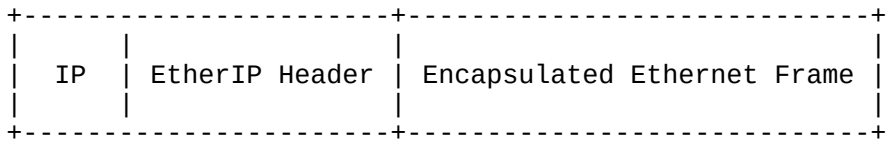

Figure 1: EtherIP Datagram Description

The 16-bit EtherIP header field consists of two parts: a 4-bit version field that identifies the EtherIP protocol version and a 12-bit field reserved for future use. The value of the version field MUST be 3 (three, '0011' binary). The value of the reserved field MUST be 0 (zero). Earlier versions of this protocol used an 8-bit header field. The Xerox Ethernet Tunnel (XET) employed the 8-bit header. The 16-bit header field provides memory alignment advantages in some implementation environments.

In summary, the EtherIP Header has two fields:

Bits 0-3: Protocol version
Bits 4-15: Reserved for future use

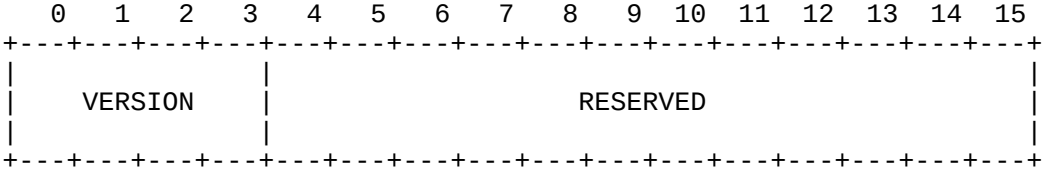

Figure 2: EtherIP Header Format (in bits)

The encapsulated Ethernet frame field contains a complete Ethernet or IEEE 802.3 frame of any type less the frame check sequence (FCS) value. The IP checksum does not provide integrity protection for the Ethernet/IEEE 802.3 frame, so some higher-layer protocol encapsulated by the Ethernet/IEEE 802.3 frame is expected to provide the integrity protection.

3. Sending Procedures

This section describes the processing to encapsulate an Ethernet or IEEE 802.3 MAC frame in an EtherIP datagram. First, the implementation determines whether the MAC frame requires tunneling. Then, if tunneling is required, the MAC frame is processed according to the steps provided in this section. Stations processing VLAN datagrams MAY need to examine the VLAN header to make appropriate tunneling decisions.

An end station that implements EtherIP may tunnel some traffic, but not all traffic. Thus, the first step in processing a MAC frame is to determine if the frame needs to be tunneled or not. If the recipient station is connected to the same LAN as the source station, then tunneling is not needed. If the network connecting the stations can route the layer three protocol, then tunneling is not needed. Other environment specific criteria MAY also be applied. If tunneling is not needed, skip all EtherIP processing and perform normal data link layer processing to transmit the MAC frame. Otherwise, follow the steps described below.

A bridge-like station promiscuously listens to all of the MAC frames on the LAN. Each MAC frame read from the LAN is examined to determine if it needs to be tunneled. If the recipient station is connected to the same LAN as the source station, then tunneling is not needed. If the destination MAC address is a router serving the LAN, then tunneling is not needed. Other environment specific criteria MAY also be applied. If tunneling is not needed, then discard the MAC frame. Otherwise, follow the steps described below.

The EtherIP encapsulation process is as follows:

1. Prepend the 16-bit EtherIP header to the MAC frame. The EtherIP Version field MUST be set to 3 (three), and the EtherIP Reserved field MUST be set to 0 (zero). The MAC frame MUST NOT include the FCS.

2. Determine the destination IP address of the remote EtherIP station. This address is usually determined from the destination MAC address.

3. Encapsulate the EtherIP datagram in an IP datagram with the destination IP address determined in the previous step, and the IPv4 Protocol field MUST be set to 97 (decimal).

4. Transmit the resulting IP datagram to the remote EtherIP station via the IP router serving the LAN.

4. Receiving Procedures

This section describes the processing to decapsulate an Ethernet or IEEE 802.3 MAC frame from an EtherIP datagram.

Since a bridge-like station promiscuously listens to all of the MAC frames on the LAN, it may need to separate the MAC frames that encapsulate IP datagrams addressed to it from MAC frames that are candidates for decapsulation. The process for identifying MAC frames that are candidates for decapsulation is as follows:

1. Perform normal data link layer processing to receive a suspected EtherIP datagram.

2. If the recipient station is connected to the same LAN as the source station, then ignore the frame. In most environments, frames with a source MAC address other than the IP router serving the LAN are ignored.

3. If the network connecting the stations can route the layer three protocol, then decapsulation is not needed, and the frame is ignored.

4. Ignore frames that do not contain an IP datagram.

5. Examine the IPv4 protocol field to confirm that the value of the field is 97 (decimal). If not, ignore the frame.

Other environment specific criteria MAY also be applied.

Upon reception of an IPv4 datagram with the Protocol field set to 97 (decimal), the MAC frame is processed as follows:

1. Examine the 16-bit EtherIP header. Confirm that the value of the version field is 3 (three), and that the value of the Reserved field is 0 (zero). Frames with other values MUST be discarded.

2. Extract the encapsulated MAC frame from the EtherIP datagram. Note that the extracted frame MUST NOT include a FCS value.

3. Perform normal data link layer processing to transmit the extracted MAC frame to the destination station on the LAN. The FCS MUST be calculated and appended to the frame as part of the data link layer transmission processing.

5. IANA Considerations

IANA has assigned IP protocol value 97 (decimal) for EtherIP. No further action or review is required.

6. Security Considerations

EtherIP can be used to enable the transfer of encrypted Ethernet or IEEE 802.3 frame payloads. In this regard, EtherIP can improve security. However, if a firewall permits EtherIP traffic to pass in and out of a protected enclave, arbitrary communications are enabled. Therefore, if a firewall is configured to permit communication using EtherIP, then additional checking of each frame is probably necessary to ensure that the security policy that the firewall is installed to enforce is not violated.

Further, the addition of EtherIP can expose a particular environment to additional security threats. Assumptions that might be appropriate when all communicating nodes are attached to one Ethernet segment or switch may no longer hold when nodes are attached to different Ethernet segments or switches are bridged together with EtherIP. It is outside the scope of this specification, which documents an existing practice, to fully analyze and review the risks of Ethernet tunneling. The IETF Pseudo-wire Emulation Working Group is doing work in this area, and this group is expected to provide information about general layering as well as specific Ethernet over IP documents. An example should make the concern clear. A number of IETF standards employ relatively weak security mechanisms when communicating nodes are expected to be connected to the same local area network. The Virtual Router Redundancy Protocol [RFC2338] is one instance. The relatively weak security mechanisms represent a greater vulnerability in an emulated Ethernet. One solution is to protect the IP datagrams that carry EtherIP with IPsec [RFC2401].

The IETF Pseudo-wire Emulation Working Group may document other security mechanisms as well.

7. Acknowledgements

This document describes a protocol that was originally designed and implemented by Xerox Special Information Systems in 1991 and 1992. An earlier version of the protocol was provided as part of the Xerox Ethernet Tunnel (XET).

8. References

[CSMA/CD] Institute of Electrical and Electronics Engineers: "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", ANSI/IEEE Std 802.3-1985, 1985.

[DIX] Digital Equipment Corporation, Intel Corporation, and Xerox Corporation: "The Ethernet -- A Local Area Network: Data Link Layer and Physical Layer (Version 2.0)", November 1982.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2119] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2338] Knight, S., Weaver, D., Whipple, D., Hinden, R., Mitzel, D., Hunt, P., Higginson, P., Shand, M. and A. Lindem, "Virtual Router Redundancy Protocol", RFC 2338, April 1998.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[SDE] Institute of Electrical and Electronics Engineers: "Interoperable LAN/MAN Security (SILS) Secure Data Exchange (SDE) (Clause 2)", IEEE Std 802.10b-1992, 1992.

[XNS] Xerox Corporation: "Internet Transport Protocols", XSIS 028112, December 1981.

[VLAN] Institute of Electrical and Electronics Engineers: "IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridge Local Area Networks", ANSI/IEEE Std 802.1Q-1998, 1998.

9. Authors' Addresses

Russell Housley
RSA Laboratories
918 Spring Knoll Drive
Herndon, VA 20170
USA

EMail: rhousley@rsasecurity.com

Scott Hollenbeck
VeriSign, Inc.
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA

EMail: shollenbeck@verisign.com

10. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.